HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200300598-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Yasushi Saito et al.	Confirmation No.: 2665
Application No.: 10/726,041	Examiner: Brent S. Stace
Filing Date: December 1, 2003	Group Art Unit: 2161

Title: Namespace Consistency for a Wide Area File System

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on December 3, 2007.

☒ The fee for filing this Appeal Brief is $510.00 (37 CFR 41.20).

☐ No Additional Fee Required.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☒(a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☒ 1st Month $120	☐ 2nd Month $460	☐ 3rd Month $1050	☐ 4th Month $1640

☒ A check in the amount of $120.00 for the extension fee is enclosed.

☐(b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $510.00. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

☒ A duplicate copy of this transmittal letter is enclosed.

☒ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:
Commissioner for Patents, Alexandria, VA 22313-1450
Date of Deposit: March 3, 2008

OR

☐ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.
Date of facsimile:
Typed Name: Derek J. Westberg
Signature:

Respectfully submitted,
Yasushi Saito et al.
By
Derek J. Westberg
Attorney/Agent for Applicant(s)
Reg No. : 40,872
Date : March 3, 2008
Telephone : (650) 968-0410

Atty. Dkt. No. 200300598-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor: Yasushi Saito	Examiner: Brent S. Stace
Serial No. 10/726,041	Art Unit: 2161
Filed: December 1, 2003	Confirmation No. 2665
Entitled: NAMESPACE CONSISTENCY FOR A WIDE AREA FILE SYSTEM	**APPEAL BRIEF**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madame:

Sir/Madame:

This is Applicant's brief on appeal from the final office action mailed on June 8, 2007.

(i) Real Party in Interest

The real party in interest is Hewlett-Packard Development Company, L.P., the assignee of record, which is a wholly-owned affiliate of Hewlett-Packard Company.

(ii) Related Appeals and Interferences

The Applicant is not aware of any appeals or interferences related to the above-identified patent application.

(iii) Status of Claims

Claims 1-15 and 56-69 are pending in this application. Claims 1-15 and 56-69 have been finally rejected and are the subject of this appeal.

(iv) Status of Amendments

All amendments have been entered.

(v) Summary of Claimed Subject Matter

Background

The Applicants' invention relates to distributed file systems and to file replication within a distributed file system. See, Applicants' specification at page 1, lines 10-13.

The Applicants' invention improves upon prior file systems. For example, NFS (Network File System) is a network file system designed for local area networks, and follows a client-server model. NFS relies on periodic polling to keep the cached data fresh. Thus, in a wide area network, NFS forces the clients to refresh data incessantly,

thus rendering NFS as very inefficient. The availability of a file is limited by the availability of the server on which the file resides. Scalability is achieved by adding more servers and more volumes; the mapping of servers-volumes-namespace is manual. See, Applicants' specification at page 1, lines 16-26.

As another example, AFS (Andrew File System) is a wide-area distributed file system that provides a unified file system under a single, global namespace. The wide-area system is organized in a number of "cells", with one cell in each physical location. Each cell comprises one or more servers. AFS utilizes persistent caching and callbacks. Write operations are flushed synchronously on the server. The "master copy" of a file resides on a single server. Thus, its availability (for open and write) depends on the availability of the server. Scalability is achieved by adding more servers and more volumes; the mapping of servers-volumes-namespace is semi-manual. See, Applicants' specification at page 1, line 27, to page 2, line 6.

As still another example, "Coda" adds to AFS two new modes of operations: "weakly connected" and "disconnected." In the case of disconnection or server failure, the client (transparently) switches to the disconnected mode and the user continues to read and write locally the cached files (that have been accessed before). However, the user cannot access files that have not been cached locally, and if the same files are updated by other clients, the changes are not visible to this client. See, Applicants' specification at page 2, lines 7-15.

Thus, the prior distributed file systems have limitations, such as in speed, availability, and network economy.

Claim 1

Applicant's claim 1 is an independent claim which is directed toward "[a] method for a wide-area file system, including a plurality of nodes storing replicas of objects, the objects being files and file directories, wherein for each replica of an object at a node, a parent directory for the object is replicated at the node." In a section entitled "Pervasive Replication," at page 11, the Applicants' specification explains that a directory is treated as a file with special contents and that a replica of a file or directory is created whenever and wherever it is accessed. The Applicants' specification explains at page 15, in the section entitled, "Name-space containment," that for every replica of a file, its parent directories are also replicated at the same node.

Claim 1 further recites that the method comprises "propagating an update to a replica of a file directory to other replicas of the file directory via a graph, wherein each replica of the file directory has edges to only a subset of the other replicas such that all the replicas of the file directory are connected via the graph." The Applicants' specification explains at least at page 22, lines 15-19, that an edge represents a connection between replicas, that updates flow along edges and that the replicas and edges comprise a strongly connected graph. The Applicants' specification at page 27, line 23, to page 28, line 28, explains the graph connections in more detail with reference to Figure 2.

Claim 1 also recites, "in response to receiving a propagated update to a replica of the file directory at a node, updating the replica for the file directory at the node." This feature is explained in the Applicants' specification at least at pages 32-33, in the section entitled, "Propagating updates."

Claim 2

Claim 2 is dependent from claim 1 and recites "wherein each replica of an object has a backpointer including an identification of a parent directory for the object and a name of the object in the parent directory." These features are explained the Applicant's specification at least at page 3, lines 29-31.

Claim 3

Claim 3 is dependent from claim 2 and recites "wherein the parent directories are modified when the backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node." Conflict resolution using the backpointers is described in the Applicants' specification at least at page 14, line 15, to page 15, line 26.

Claim 4

Claim 4 is dependent from claim 3 and recites "wherein modifying the parent directories occurs only after a delay." This feature is described in the Applicants' specification at least at page 16, line 25, to page 17, line 17.

Claim 6

Claim 6 is dependent from claim 3 and recites "wherein a modification is performed at the node and an earlier inconsistent modification is ignored." This feature is described in the Applicants' specification at least at page 38, line 20, to page 41, line 14.

Claim 9

Claim 9 is dependent from claim 8 and recites "wherein when the backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node, further comprising modifying the parent directories to be consistent with the backpointer." Conflict resolution using the backpointers is described in the Applicants' specification at least at page 14, line 15, to page 15, line 26.

Claim 10

Claim 10 is dependent from claim 1 and recites "wherein the replicas of the file directory include core replicas and non-core replicas, the parent directory for the file directory having edges only to the core replicas of the file directory and each core replica of the file directory having edges to one or more of the non-core replicas of the file directory." The Applicant's specification explains the these two types of replicas at least at page 26, line 16, to page 28, line 16.

Claim 11

Claim 11 is dependent from claim 10 and recites "wherein in response to a user accessing an object at a node when no replica of the object exists at the node, the method further comprises steps of forming a non-core replica of the parent directory for the object at the node and forming a non-core replica of the object at the node." These features are described in the Applicants' specification at least at page 34, line 9, to page 28, line 8.

Claim 12

Claim 12 is dependent from claim 10 and recites "wherein a minimum number of core replicas are maintained according to a minimum replication factor." This feature is described in the Applicants' specification at least at page 26, line 31, to page 27, line 5.

Claim 13

Claim 13 is dependent from claim 1 and recites "wherein a replica of an object is deleted by marking the replica as invalid." This feature is described in the Applicants' specification at least at page 19, line 26, to page 30, line 4.

Claim 14

Claim 14 is dependent from claim 1 and recites "wherein said marking the replica as invalid comprises removing the backpointer for the replica." This feature is described in the Applicants' specification at least at page 44, line 19, to page 45, line 18.

Claim 56

Applicant's claim 56 is an independent claim which is directed toward a system including "a plurality of nodes that store replicas of objects, the objects being files and file directories." In a section entitled "Pervasive Replication," at page 11, the Applicants' specification explains that a directory is treated as a file with special contents and that a replica of a file or directory is created whenever and wherever it is accessed.

Claim 56 further recites "wherein for each replica of an object at a node, the node stores a replica of a parent directory for the object and a backpointer having an identification of the parent directory for the object." The Applicants' specification explains at page 15, in the section entitled, "Name-space containment," that for every replica of a file, its parent directories are also replicated at the same node. In addition, the Applicants specification explains at page 3, lines 29-31, that each replica includes a backpointer having an identification of a parent directory for the file and a name of the file within the parent directory. And, as explained at page 10, lines 20-22, to page 22, the backpointer authoritatively defines a file' location in the file system's namespace. Backpointers are also described at page 14, line 15, to page 15, line 26.

Claim 56 further recites "wherein each replica of a file directory has edges to only a subset of the other replicas of the file directory such that all the replicas of the file directory are connected via the graph and the nodes are configured to propagate updates to replicas of each file directory to other replicas of the file directory via the graph." The Applicants' specification explains at least at page 22, lines 15-19, that an edge represents a connection between replicas, that updates flow along edges and that the replicas and edges comprise a strongly connected graph. The Applicants' specification at page 27, line 23, to page 28, line 28, explains the graph connections in more detail with reference to Figure 2.

Claim 57

Claim 57 is dependent from claim 56 and recites "wherein in response to receiving a propagated update to a replica of the file directory at a node, the node updates

the parent directories for the file at the node." This feature is explained in the Applicants' specification at least at pages 32-33, in the section entitled, "Propagating updates."

Claim 58

Claim 58 is dependent from claim 56 and recites "wherein when a backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node, the node modifies the parent directories to be consistent with the backpointer." Conflict resolution using the backpointers is described in the Applicants' specification at least at page 14, line 15, to page 15, line 26.

Claim 59

Claim 59 is dependent from claim 58 and recites "wherein the node modifies the parent directories to be consistent with the backpointer only after a delay." This feature is described in the Applicants' specification at least at page 16, line 25, to page 17, line 17.

Claim 61

Claim 61 is dependent from claim 58 and recites "wherein a modification is performed at the node and an earlier inconsistent modification is ignored." This feature is described in the Applicants' specification at least at page 38, line 20, to page 41, line 14.

Claim 64

Claim 64 is dependent from claim 62 and recites "wherein when a backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node, the node modifies the parent directories to be consistent with the backpointer." The Conflict resolution using the backpointers is described in the Applicants' specification at least at page 14, line 15, to page 15, line 26.

Claim 65

Claim 65 is dependent from claim 56 and recites "wherein the replicas of the file directory include core replicas and non-core replicas, the parent directory for the file directory having edges only to the core replicas of the file directory and each core replica of the file directory having edges to one or more of the non-core replicas of the file directory." The Applicant's specification explains the these two types of replicas at least at page 26, line 16, to page 28, line 16.

Claim 66

Claim 66 is dependent from claim 56 and recites "wherein in response to a user accessing an object at a node when no replica of the object exists at the node, a non-core replica of the object and a non-core replica of the parent directory for the object are formed at the node." These features are described in the Applicants' specification at least at page 34, line 9, to page 28, line 8.

Claim 67

Claim 67 is dependent from claim 56 and recites "wherein a replica of an object is deleted by marking the replica as invalid." This feature is described in the Applicants' specification at least at page 19, line 26, to page 30, line 4.

Claim 68

Claim 68 is dependent from claim 67 and recites "wherein said marking the replica as invalid comprises removing the backpointer for the replica." This feature is described in the Applicants' specification at least at page 44, line 19, to page 45, line 18.

(vi) Grounds of Rejection to be Reviewed on Appeal

Whether the drawings are properly objected to under 37 C.F.R. 1.84(p)(5) as including the reference character "405" which is allegedly not mentioned in the description.

Whether claims 11 and 59-69 are unpatentable under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite.

Whether claims 1, 10 and 11 are unpatentable under 35 U.S.C. § 103 as allegedly being obvious over "Replication in Ficus Distributed File Systems," by Popek et al. (hereinafter "Popek") in view of U.S. Patent Publication No. 2001/0044879 by Moulton, et al. (hereinafter, "Moulton").

Whether claims 2-9 and 56-66 are unpatentable under 35 U.S.C. § 103 as allegedly being obvious over Popek in view of Moulton and further in view of

"Designing a Robust Namespace for Distributed File Services" by Zhang et al. (hereinafter, "Zhang").

Whether claim 12 is unpatentable under 35 U.S.C. § 103 as allegedly being obvious over Popek in view of Moulton and further in view of "The Costs and Limits of Availability for Replicated Services" by Yu et al. (hereinafter, "Yu").

Whether claims 13 and 15 are unpatentable under 35 U.S.C. § 103 as allegedly being obvious over Popek in view of Moulton and further in view of U.S. Patent Publication No. 2002/0107835 by Coram et al. (hereinafter, "Coram").

Whether claims 14 and 67-69 are unpatentable under 35 U.S.C. § 103 as allegedly being obvious over Popek, Moulton and Coram, and further in view of Zhang.

(vii) Argument

a. Objection to the Drawings

The drawings are objected to under 37 C.F.R. 1.84(p)(5) as including the reference character "405" which is allegedly not mentioned in the description. The Applicants respectfully disagree. The reference character "405" is mentioned in the description at least in the paragraph which was replaced by amendment dated May 7, 2007. The reference character "405" appears in line 8.

b. Rejections under 35 U.S.C. § 112, Second Paragraph

Claims 11 and 66

Claims 11 and 66 were rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite. The final rejection mailed on July 31, 2007, states essentially

that the recitation, "in response to a user accessing an object at a node when no replica of the object exists at the node" is indefinite because "[a]n object cannot be accessed when it doesn't' exist." The final rejection mailed on July 31, 2007, further indicates that the Applicants intended to claim "in response to a user attempting to access an object at a node...". See, final rejection mailed on July 31, 2007, at pages 4-5, (emphasis in original).

The Applicants respectfully disagree with the rejection. Regarding definiteness of claim language, the Manual of Patent Examining Procedure (MPEP) states as follows:

> The essential inquiry pertaining to this requirement is whether the claims set out and circumscribe a particular subject matter with a reasonable degree of clarity and particularity. Definiteness of claim language must be analyzed, not in a vacuum, but in light of:
> (A) The content of the particular application disclosure;
> (B) The teachings of the prior art; and
> (C) The claim interpretation that would be given by one possessing the ordinary level of skill in the pertinent art at the time the invention was made.

MPEP at Section 2173.02 (Sept. 2007). Thus, the Applicants respectfully submit that claims 11 and 66 should be considered for definiteness in their entireties, and not based solely on the fragment of the claims which is quoted in the final rejection. When claims 11 and 66 are considered in their entireties, it can be seen that they also recite forming a replica of the parent directory for the object at the node and forming a non-core replica of the object at the node. This feature is explained in a section entitled, "Creating a replica," which begins at page 34 of the Applicants' specification. The Applicants respectfully submit that these claims are sufficiently clear when considered as a whole and when considered in light of the Applicants' specification.

In view of the above, the Applicants respectfully request reversal of the rejection. Alternatively, the Applicants hereby offer to amend claims 11 and 66 as suggested in the final rejection.

Claims 59-69

Claims 59-69 rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite. The final rejection mailed on July 31, 2007, states essentially that claim 56 recites the limitation, "the graph" in lines 9 and 10 and that there is insufficient antecedent basis for this claim limitation. Claims 57-69 are rejected as being dependent from claim 57.

The Applicants respectfully disagree with the rejection. Regarding antecedent basis, the Manual of Patent Examining Procedure (MPEP) states as follows:

> A claim is indefinite when it contains words or phrases whose meaning is unclear. The lack of clarity could arise where a claim refers to "said lever" or "the lever," where the claim contains no earlier recitation or limitation of a lever and where it would be unclear as to what element the limitation was making reference. Similarly, if two different levers are recited earlier in the claim, the recitation of "said lever" in the same or subsequent claim would be unclear where it is uncertain which of the two levers was intended. A claim which refers to "said aluminum lever," but recites only "a lever" earlier in the claim, is indefinite because it is uncertain as to the lever to which reference is made. Obviously, however, the failure to provide explicit antecedent basis for terms does not always render a claim indefinite. If the scope of a claim would be reasonably ascertainable by those skilled in the art, then the claim is not indefinite. >*Energizer Holdings Inc. v. Int'l Trade Comm'n, 435 F.3d 1366, 77 USPQ2d 1625 (Fed. Cir. 2006)(holding that "anode gel" provided by implication the antecedent basis for "zinc anode");*< *Ex parte Porter*, 25 USPQ2d 1144, 1145 (Bd. Pat. App. & Inter. 1992) ("controlled stream of fluid" provided reasonable antecedent basis for "the controlled fluid"). Inherent components of elements recited have antecedent basis in the recitation of the components themselves. For example, the limitation "the outer surface of said sphere" would not require an antecedent recitation that the sphere has an outer surface. See *Bose Corp. v. JBL, Inc.*, 274 F.3d 1354, 1359, 61 USPQ2d 1216, 1218-19 (Fed. Cir 2001) (holding that recitation of "an ellipse" provided antecedent basis for "an ellipse having a

> major diameter" because "[t]here can be no dispute that mathematically an inherent characteristic of an ellipse is a major diameter").

MPEP at Section 2173.05(e) (Sept. 2007). Here, claim 56 recites, "wherein each replica of a file directory has edges to only a subset of the other replicas of the file directory such that all the replicas of the file directory are connected via the graph and the nodes are configured to propagate updates to replicas of each file directory to other replicas of the file directory via the graph." Thus, by clear implication within the claim itself, "the graph" is composed of the previously-recited "edges."

In view of the above, the Applicants respectfully request reversal of the rejection. Alternatively, the Applicants hereby offer to amend claim 56 to replace "the graph" with "a graph."

b. *Rejections under 35 U.S.C. § 103 in view of Popek and Moulton*

Claim 1

Claim 1 was rejected under 35 U.S.C. § 103 as allegedly being obvious over Popek in view Moulton. Particularly, the final rejection alleges that Popek discloses all of the limitations of claim 1 in three paragraphs (on page 21, the first paragraph under the heading, "5 The Ficus Project" and on page 22, the first and third paragraphs under the heading, "3 The Optimistic Model"), except that Popek does not teach "via a graph, where each replica of the file directory has edges to only a subset of the other replicas such that all the replicas of the file directory are connected via the graph." See, final rejection mailed on July 31, 2007, at pages 6-7. The final rejection further alleges that Moulton teaches this limitation in Figure 1 and that "[i]t would have been obvious to one of ordinary skill in the art at the time of the invention having the teachings of Moulton

and Popek before him/her to combine Moulton with Popek because both inventions are directed towards replicating files across computers." See, final rejection mailed on July 31, 2007, at pages 6-7. The final rejection also alleges that "[i]t would have been obvious to one of ordinary skill in the art at the time of the invention having the teachings of Moulton and Popek before him/her to take the graph from Moulton and install it into the invention of Popek, thereby offering the obvious advantage of having an efficient means of determining how the replicas replicate (Moulton, at paragraphs [0032] and [0036])." See, final rejection mailed on July 31, 2007, at page 7.

The Applicants respectfully disagree with the rejection. "Under §103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented." *Graham* v. *John Deere Co. of Kansas City*, 383 U.S. 1, 17-18 (1966). *See also*, *KSR Intl. Co. v. Teleflex Inc.*, 550 U.S. ___ (2007). Moreover, in order to properly reject a patent claim under 35 U.S.C. § 103, the claimed subject matter must be considered as a whole. *See* 35 U.S.C. § 103.

When these principles are followed, it is apparent that the Applicants' claimed invention is not obvious in view of the cited references. Particularly, the Applicants respectfully submit that it would not have been obvious to combine Moulton with Popek

in the manner suggested in the final rejection. And, even if combined, the Popek and Moulton references to do not disclose all of the limitations of Applicants' claim 1.

Popek et al. describes a distributed file system, referred to as "Ficus." See title of Popek et al. Popek states that "[o]ne of the primary difficulties with replication services in a distributed environment concerns what to do about an update. The issue is one of mutual consistency; keeping the multiple copies of an object consistent with one another." See, Popek at page 21, first paragraph under the heading, "3 The Optimistic Model." Popek further discusses:

> It is our position that the large scale distributed filing arena requires a different approach to concurrency control from that employed in distributed databases. In the database view, one guarantees that either all objects updated atomically as a unit or none are, in a manner that assures serializable schedules. By contrast, serializability and atomic transactions are not provided by today's local file systems and are not required, in general, in their large scale distributed descendants. We argue that one should not even apply transaction models to the collection of copies of an individual object, as that means update (and even, in some cases, reading) is impossible if any particular copy is missing. However, distributed file systems should provide the hooks whereby higher levels of consistency (up to and including serializability) can be provided to clients who require it.

Popek at page 23, second paragraph under the headling, "Conclusions and Outstanding Issues." Thus, Popek clearly discusses that concepts taken from the field of databases and transaction-based data storage systems are not applicable to distributed file systems, such as the one described by Popek. However, Moulton describes just such a system. This is apparent because Moulton is not directed toward a distributed file system, but is instead directed toward a "System and Method for *Distributed Data Storage*." See, title of Moulton. More particularly, in accordance with Moulton, data is stored across multiple nodes in a fault-tolerant fashion in a manner similar to RAID storage systems. See, Moulton at paragraphs [0039] and [0067]-[0087]; and Figures 7A-

F. Thus, a single data element is broken into multiple pieces with each piece, along with an error correction code, being stored at a different node such that the original data element can be reconstructed in the event of a failure of one of the nodes. See, Moulton at paragraphs [0079]-[0080]. These schemes require that the data being stored across the multiple nodes is coherent with the data at the others nodes (i.e. the pieces of the data element are updated atomically as a unit); otherwise, it would not be possible to recover the data in the event of a loss of data. See, Moulton at paragraph [0052].

Therefore, because Popek teaches that concepts taken from the field of databases and transaction-based data storage systems are not applicable to distributed file systems, and Moulton describes just such a system, Popek teaches away from making a combination of Moulton with Popek. For at least this reason, the Applicants respectfully request reversal of the rejection of claim 1.

Moreover, the Applicants respectfully disagree with the motivations for making such a combination which are alleged in the final rejection. More particularly, the final rejection alleges that it would have been obvious to combine Popek and Moulton since both are directed toward replicating files "across" computers. See, final rejection mailed on July 31, 2007, at page 7. The Applicants respectfully disagree. As explained above, Moulton is directed toward breaking a single data element into pieces and storing the pieces "across" multiple nodes, whereas, Popek is directed toward replication in the Ficus distributed file system. See, Title of Popek. Therefore, because Popek does not store files "across" nodes in the manner of Moulton, the Applicants respectfully disagree with this alleged motivation.

The final rejection also alleges that a person of ordinary skill in the art would have been motivated to combine Popek with Moulton to have "an efficient means of determining how the replicas replicate (Moulton, at paragraphs [0032] and [0036])." See, final rejection mailed on July 31, 2007, at page 7. The Applicants respectfully disagree. At paragraph [0032] Moulton describes the network shown in Figure 1 of Moulton as being the Internet. And, at paragraph [0036] Moulton explains that storage nodes 105 and storage management mechanisms 106 are merged in the sense that both are implemented at each node 105/106. Thus, the Applicants respectfully submit that these portions of Moulton are unrelated to the alleged motivation of having an "efficient means of determining how the replicas replicate."

For at least these reasons, the Applicants respectfully submit that it would not have been obvious to combine Popek with Moulton in the manner suggested in the final rejection.

Moreover, even if considered in combination, the Applicants respectfully submit that the Popek and Moulton references to do not disclose all of the limitations of Applicants' claim 1. More particularly, claim 1 recites "a plurality of nodes storing replicas of objects, the objects being files and file directories, wherein for each replica of an object at a node, a parent directory for the object is replicated at the node." Therefore, claim 1 requires that for each replica of an object at a node, which includes files and file directories, a parent directory for the object is replicated at the node. As discussed in the Applicants' specification at pages 15-16, replication of the parent directory is referred to by the Applicants' as "namespace containment" and is a mechanism by which availability within the distributed file system is improved and by which administration of the system

is eased. The Applicants respectfully submit that neither Popek, nor Moulton, suggests or discloses such a feature. Therefore, the Applicants respectfully submit that this is another reason why claim 1 is allowable over Popek and Moulton.

Further, claim 1 recites "propagating an update to a replica of a file directory to other replicas of the file directory via a graph, wherein each replica of the file directory has edges to only a subset of the other replicas such that all the replicas of the file directory are connected via the graph." Accordingly, claim 1 requires that replicas of file directories are connected via a graph such that each replica of the file directory has edges to only a subset of the other replicas such that all the replicas of the file directory are connected via the graph. Therefore, it necessarily follows from claim 1 that some replicas of file directories are directly connected by an edge, whereas, other replicas are only indirectly connected via one or more intermediate nodes. Updates are propagated along the graph. Therefore, it also necessarily follows from claim 1 some updates to the replicas of the file directories are propagated directly from one node to another, whereas, other updates to the replicas of the file directories are propagated via the one or more intermediate nodes. The Applicants respectfully submit that neither Popek, nor Moulton suggests or discloses these features. Particularly, Popek does not discuss connections between replicas of file directories at all. Therefore, Popek cannot disclose the particular manner of connecting file directories and the particular manner of propagating updates to the file directories, which are recited in Applicants' claim 1. And, while Moulton discusses the Internet, there is no teaching or suggestion in Moulton that file directories might be connected in the particular manner recited by Applicants' claim 1, nor that updates to the file directories may be propagated in the particular manner recited by

Applicants' claim 1. Therefore, the Applicants respectfully submit that this is yet another reason why claim 1 is allowable over Popek and Moulton.

In view of the above, the Applicants respectfully request reversal of the rejection of claim 1.

Claim 10

Claim 10 is dependent from claim 1 and recites "wherein the replicas of the file directory include core replicas and non-core replicas, the parent directory for the file directory having edges only to the core replicas of the file directory and each core replica of the file directory having edges to one or more of the non-core replicas of the file directory." Thus, claim 10 further defines the manner in which the replicas of the file directories are connected.

The final rejection alleges that Moulton discloses these features at paragraphs [0028] and [0078]. See, final rejection mailed on July 31, 2007, at pages 7-8 and 13. The Applicants respectfully disagree with the rejection. As explained above in connection with claim 1, Popek does not discuss connections between replicas of file directories at all. Therefore, Popek cannot disclose the particular manner of connecting replicas of file directories and the particular manner of propagating updates to the replicas of file directories, which are recited in Applicants' claim 10. And, while Moulton discusses the Internet, there is no teaching or suggestion in Moulton that file directories might be connected in the particular manner recited by Applicants' claim 10. Therefore, the Applicants respectfully submit that this is yet another reason why claim 10 is allowable over Popek and Moulton.

Claim 11

Claim 11 is dependent from claim 1 and recites "wherein in response to a user accessing an object at a node when no replica of the object exists at the node, the method further comprises steps of forming a non-core replica of the parent directory for the object at the node and forming a non-core replica of the object at the node."

The final rejection alleges that Moulton discloses these features at paragraph [0078]. See, final rejection mailed on July 31, 2007, at pages 8 and 13. The Applicants respectfully disagree with the rejection. Paragraph [0078] of Moulton discusses data mirroring and, in particular, that "[r]ead operations attempt to first read data from one of the nodes, and if that node is unavailable, a read from the mirror node is attempted." However, this does not suggest or disclose forming such mirror data at the node at which the read was first attempted. Moreover, this does not suggest or disclose forming "a non-core replica of the parent directory for the object at the node," as is recited by Applicants' claim 11. Popek does not disclose the features of claim 11 either. Accordingly, these are additional reasons why claim 11 is allowable over Popek and Moulton.

c. Rejections under 35 U.S.C. § 103 in view of Popek, Moulton and Zhang

Claim 2

Claim 2 is dependent from claim 1 and recites "wherein each replica of an object has a backpointer including an identification of a parent directory for the object and a name of the object in the parent directory." Regarding claim 2, the final rejection mailed on July 31, 2007, states essentially that Popek and Moulton do not disclose these

limitations, but that Zhang discloses them. The final rejection alleges that "Zhang's invention would have been expected to work successfully with Popek (as modified by Moulton)'s invention [sic] because both inventions use file replication." See, final rejection mailed on July 31, 2007, at pages 8-9.

The Applicants respectfully disagree with the rejection. Claim 2 is allowable at least because it depends from an allowable base claim 1. Further, as explained by the Applicants in response to a prior rejection, Zhang et al. is directed toward distributed file services. See Abstract and Title of Zhang et al. It is important to note that Zhang does not discuss file replication except to note (in a footnote) that they consider that replicas of an object correspond to one logical object. Thus, Zhang et al. treats any replicas of an object as a single logical object. From this, it is clear that Zhang et al. does not teach or suggest the use of a backpointer in such a system in which files and file directories are replicated. Zhang also does not suggest or disclose how such a backpointer might be employed in such a system in which files and file directories are replicated.

In view of the above, the Applicants respectfully submit that claim 2 is allowable.

Claim 3

Claim 3 is dependent from claim 2 and recites "wherein the parent directories are modified when the backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node." The final rejection mailed on July 31, 2007, states essentially that Popek and Moulton do not disclose these limitations, but that Zhang discloses them at pages 2-3, under the heading, "2 Problem Abstraction." See, final rejection mailed on July 31, 2007, at pages 9-10.

The Applicants respectfully disagree with the rejection. Claim 3 is allowable at least because it depends from an allowable claims 1 and 2. Further, as explained previously, Zhang et al. is directed toward distributed file services. See Abstract and Title of Zhang et al. Because Zhang et al. treats any replicas of an object as a single logical object, Zhang does not suggest any actions that might be taken to detect whether there are inconsistencies among replicas of file or file directory, nor any actions that might be taken is response to detection of such an inconsistency.

In view of the above, the Applicants respectfully submit that claim 3 is allowable.

Claim 4

Claim 4 is dependent from claim 3 and recites "wherein modifying the parent directories occurs only after a delay." The final rejection mailed on July 31, 2007, states essentially that Popek and Moulton do not disclose these limitations, but that Zhang discloses them at pages 2-3, under the heading, "2 Problem Abstraction." See, final rejection mailed on July 31, 2007, at page 10.

The Applicants respectfully disagree with the rejection. Zhang does not suggest or disclose such a delay. Therefore, this is another reason why the Applicants respectfully submit that claim 4 is allowable.

Claim 5

Claim 5 stands or falls together with claim 3 from which it depends.

Claim 6

Claim 6 is dependent from claim 3 and recites, "wherein a modification is performed at the node and an earlier inconsistent modification is ignored." The final rejection mailed on July 31, 2007, alleges that Popek discloses these limitations at page 21, in the middle paragraph in the second column, and that Zhang discloses these limitations at page 3, in the paragraph above the heading, "3 System Model and Failure Assumptions." See, final rejection mailed on July 31, 2007, at page 10.

The Applicants respectfully disagree with the rejection. While the cited portion of Popek refers to "conflicting updates" and the cited portion of Zhang refers to a "possible inconsistency" caused by a failure, neither suggests or discloses the particular limitations recited by Applicants' claim 6. Therefore, this is another reason why the Applicants respectfully submit that claim 6 is allowable.

Claims 7-8

Claims 7-8 stand or fall together with claim 2 from which they depend.

Claim 9

Claim 9 is dependent from claim 8 and recites "wherein when the backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node, further comprising modifying the parent directories to be consistent with the backpointer." The final rejection mailed on July 31, 2007, states essentially that Popek and Moulton do not disclose these limitations, but that Zhang

discloses them at pages 2-3, under the heading, "2 Problem Abstraction," and in the paragraphs under Table 2. See, final rejection mailed on July 31, 2007, at page 11.

The Applicants respectfully disagree with the rejection. As explained previously, Zhang et al. is directed toward distributed file services. See Abstract and Title of Zhang et al. Because Zhang et al. treats any replicas of an object as a single logical object, Zhang does not suggest any actions that might be taken to detect whether there are inconsistencies among replicas of file or file directory, nor any actions that might be taken is response to detection of such an inconsistency.

In view of the above, the Applicants respectfully submit that claim 9 is allowable.

Claim 56

Claim 56 is an independent claim which was rejected using reasoning similar to that of claim 2. See, final rejection mailed on July 31, 2007, at pages 11-13. The Applicants respectfully disagree with the rejection. Similarly to Applicants' claim 1, claim 56 recites "wherein each replica of a file directory has edges to only a subset of the other replicas of the file directory such that all the replicas of the file directory are connected via the graph and the nodes are configured to propagate updates to replicas of each file directory to other replicas of the file directory via the graph." As explained above in connection with Applicants' claim 1, because Popek teaches that concepts taken from the field of databases and transaction-based data storage systems are not applicable to distributed file systems, and Moulton describes just such a system, Popek teaches away from making a combination of Moulton with Popek. Zhang does not cure this deficiency

in Popek and Moulton. For at least this reason, the Applicants respectfully request reversal of the rejection of claim 56.

Further, the Applicants respectfully disagree with the motivations alleged in the final rejection for making the combination of Popek, Moulton and Zhang. Particularly, the final rejection alleges that "Zhang and Moulton's inventions would have been expected to successfully work well with Popek because the inventions are directed toward distributed file replication." See, final rejection mailed on July 31, 2007, at page 12. As explained previously, however, Zhang et al. is directed toward distributed file services. See Abstract and Title of Zhang et al. It is important to note that Zhang does not discuss file replication except to note (in a footnote) that they consider that replicas of an object correspond to one logical object. Thus, Zhang et al. treats any replicas of an object as a single logical object. From this, it is clear that Zhang et al. does not teach or suggest the use of a backpointer in such a system in which files and file directories are replicated. Zhang also does not suggest or disclose how such a backpointer might be employed in such a system in which files and file directories are replicated.

The final rejection also alleges that a person of ordinary skill in the art would have been motivated to combine Popek with Moulton to have "an efficient means of determining how the replicas replicate (Moulton, at paragraphs [0032] and [0036])." See, final rejection mailed on July 31, 2007, at page 13. The Applicants respectfully disagree. At paragraph [0032] Moulton describes the network shown in Figure 1 of Moulton as being the Internet. And, at paragraph [0036] Moulton explains that storage nodes 105 and storage management mechanisms 106 are merged in the sense that both are implemented at each node 105/106. Thus, the Applicants respectfully submit that these

portions of Moulton are unrelated to the alleged motivation of having an "efficient means of determining how the replicas replicate."

Further, claim 56 recites "wherein each replica of a file directory has edges to only a subset of the other replicas of the file directory such that all the replicas of the file directory are connected via the graph and the nodes are configured to propagate updates to replicas of each file directory to other replicas of the file directory via the graph." Accordingly, claim 56 requires that replicas of file directories are connected via a graph such that each replica of the file directory has edges to only a subset of the other replicas such that all the replicas of the file directory are connected via the graph. Therefore, it necessarily follows from claim 56 that some replicas of file directories are directly connected by an edge, whereas, other replicas are only indirectly connected via one or more intermediate nodes. Updates are propagated along the graph. Therefore, it also necessarily follows from claim 56 that some updates to the replicas of the file directories are propagated directly from one node to another, whereas, other updates to the replicas of the file directories are propagated via the one or more intermediate nodes. The Applicants respectfully submit that neither Popek, Moulton, nor Zhang suggests or discloses such a feature. Particularly, Popek does not discuss connections between replicas of file directories at all. Therefore, Popek cannot disclose the particular manner of connecting file directories and the particular manner of propagating updates to the file directories, which are recited in Applicants' claim 56. And, while Moulton discusses the Internet, there is no teaching or suggestion in Moulton that file directories might be connected in the particular manner recited by Applicants' claim 56. Zhang et al. treats any replicas of an object as a single logical object. Therefore, the Applicants respectfully

submit that this is yet another reason why claim 56 is allowable over Popek, Moulton and Zhang.

Claim 56 also recites "wherein for each replica of an object at a node, the node stores a replica of a parent directory for the object and a backpointer having an identification of the parent directory for the object." Regarding these limitations, the final rejection mailed on July 31, 2007, states essentially that Popek and Moulton do not disclose these limitations, but that Zhang discloses them. The final rejection alleges that Zhang and Moulton's inventions would have been expected to successfully work well with Popek because the inventions are directed toward distributed file replication." See, final rejection mailed on July 31, 2007, at page 12. The Applicants respectfully disagree. Zhang et al. is directed toward distributed file services. See Abstract and Title of Zhang et al. It is important to note that Zhang does not discuss file replication except to note (in a footnote) that they consider that replicas of an object correspond to one logical object. Thus, Zhang et al. treats any replicas of an object as a single logical object. From this, it is clear that Zhang et al. does not teach or suggest the use of a backpointer in such asystem in which files and file directories are replicated. Zhang also does not suggest or disclose how such a backpointer might be employed in such a system in which files and file directories are replicated. Therefore, this is another reason why claim 56 is allowable.

In view of the above, the Applicants respectfully submit that claim 56 is allowable.

Claim 57

Claim 57 is dependent from claim 56 and recites, "wherein in response to receiving a propagated update to a replica of the file directory at a node, the node updates the parent directories for the file at the node." The final rejection alleges that Popek teaches these limitations at page 21, in the third and fourth paragraphs under the heading, "3 The Optimistic Model" and page 22, in the first through third paragraphs under the heading, "5 The Ficus Project."

The Applicants respectfully disagree with the rejection. These portions of Popek do not discuss updating parent directories. Accordingly, this portions of Popek cannot disclose the specific limitations of Applicants' claim 57 which relate to updating parent directories.

The Applicants respectfully submit that this is another reason why claim 57 is allowable.

Claim 58

Claim 58 is dependent from claim 56 and recites "wherein when a backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node, the node modifies the parent directories to be consistent with the backpointer." The final rejection mailed on July 31, 2007, states essentially that Popek and Moulton do not disclose these limitations, but that Zhang discloses them at pages 2-3, under the heading, "2 Problem Abstraction." See, final rejection mailed on July 31, 2007, at pages 9-10 and 13.

The Applicants respectfully disagree with the rejection. Claim 58 is allowable at least because it depends from an allowable claim 56. Further, as explained previously, Zhang et al. is directed toward distributed file services. See Abstract and Title of Zhang et al. Because Zhang et al. treats any replicas of an object as a single logical object, Zhang does not suggest any actions that might be taken to detect whether there are inconsistencies among replicas of file or file directory, nor any actions that might be taken is response to detection of such an inconsistency.

In view of the above, the Applicants respectfully submit that claim 58 is allowable.

Claim 59

Claim 59 is dependent from claim 58 and recites "wherein the node modifies the parent directories to be consistent with the backpointer only after a delay." The final rejection mailed on July 31, 2007, states essentially that Popek and Moulton do not disclose these limitations, but that Zhang discloses them at pages 2-3, under the heading, "2 Problem Abstraction." See, final rejection mailed on July 31, 2007, at pages 10 and 13.

The Applicants respectfully disagree with the rejection. Zhang does not suggest or disclose such a delay. Therefore, this is another reason why the Applicants respectfully submit that claim 59 is allowable.

Claim 60

Claim 60 stands or falls together with claim 58 from which it depends.

Claim 61

Claim 61 is dependent from claim 58 and recites, "wherein a modification is performed at the node and an earlier inconsistent modification is ignored." The final rejection mailed on July 31, 2007, alleges that Popek discloses these limitations at page 21, in the middle paragraph in the second column, and that Zhang discloses these limitations at page 3, in the paragraph above the heading, "3 System Model and Failure Assumptions." See, final rejection mailed on July 31, 2007, at pages 10 and 13.

The Applicants respectfully disagree with the rejection. While the cited portion of Popek refers to "conflicting updates" and the cited portion of Zhang refers to a "possible inconsistency" caused by a failure, neither suggests or discloses the particular limitations recited by Applicants' claim 61. Therefore, this is another reason why the Applicants respectfully submit that claim 61 is allowable.

Claims 62-63

Claims 62-63 stand or fall together with claim 2 from which they depend.

Claim 64

Claim 64 is dependent from claim 62 and recites "wherein when a backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node, the node modifies the parent directories to be consistent with the backpointer." The final rejection mailed on July 31, 2007, states essentially that Popek and Moulton do not disclose these limitations, but that Zhang discloses them at pages 2-3,

under the heading, "2 Problem Abstraction," and in the paragraphs under Table 2. See, final rejection mailed on July 31, 2007, at pages 11 and 13.

The Applicants respectfully disagree with the rejection. As explained previously, Zhang et al. is directed toward distributed file services. See Abstract and Title of Zhang et al. Because Zhang et al. treats any replicas of an object as a single logical object, Zhang does not suggest any actions that might be taken to detect whether there are inconsistencies among replicas of file or file directory, nor any actions that might be taken is response to detection of such an inconsistency.

In view of the above, the Applicants respectfully submit that claim 64 is allowable.

Claim 65

Claim 65 is dependent from claim 56 and recites "wherein the replicas of the file directory include core replicas and non-core replicas, the parent directory for the file directory having edges only to the core replicas of the file directory and each core replica of the file directory having edges to one or more of the non-core replicas of the file directory." Thus, claim 65 further defines the manner in which the replicas of the file directories are connected.

The final rejection alleges that Moulton discloses these features at paragraphs [0028] and [0078]. See, final rejection mailed on July 31, 2007, at pages 7-8 and 13. The Applicants respectfully disagree with the rejection. As explained above in connection with claims 1 and 56, Popek does not discuss connections between replicas of file directories at all. Therefore, Popek cannot disclose the particular manner of

connecting replicas of file directories and the particular manner of propagating updates to the replicas of file directories, which are recited in Applicants' claim 65. And, while Moulton discusses the Internet, there is no teaching or suggestion in Moulton that file directories might be connected in the particular manner recited by Applicants' claim 65. Zhang does not disclose the features of claim 65 either. Therefore, the Applicants respectfully submit that this is yet another reason why claim 65 is allowable.

Claim 66

Claim 66 is dependent from claim 56 and recites "wherein in response to a user accessing an object at a node when no replica of the object exists at the node, a non-core replica of the object and a non-core replica of the parent directory for the object are formed at the node."

The final rejection alleges that Moulton discloses these features at paragraph [0078]. See, final rejection mailed on July 31, 2007, at pages 8 and 13. The Applicants respectfully disagree with the rejection. Paragraph [0078] of Moulton discusses data mirroring and, in particular, that "[r]ead operations attempt to first read data from one of the nodes, and if that node is unavailable, a read from the mirror node is attempted." However, this does not suggest or disclose forming any mirror data at the node at which the read was first attempted. Moreover, this does not suggest or disclose forming "a non-core replica of the parent directory for the object at the node," as is recited by Applicants' claim 11. Popek and Zhang do not disclose the features of claim 66 either. Accordingly, these are additional reasons why claim 66 is allowable.

d. *Rejections under 35 U.S.C. § 103 in view of Popek, Moulton and Yu*

Claim 12

Claim 12 is dependent from claim 10 and recites "wherein a minimum number of core replicas are maintained according to a minimum replication factor." The final rejection alleges that Yu discloses this feature at: page 1, last paragraph; page 12, Figs. 11 and 12; page 12, last paragraph in the section entitled, "5.3 Effects of Replication Scale," and in Popek at page 21, in the third paragraph under the heading "3 Optimistic Model."

The Applicants respectfully disagree with the rejection. Claim 12 further defines the manner in which the replicas of the file directories are connected by requiring a minimum number of the recited "core replicas." As explained above in connection with claims 1 and 10, Popek and Moulton do not disclose the particular manner of connecting replicas of file directories and the particular manner of propagating updates to the replicas of the file directories, which are recited in Applicants' claim 10. Thus, Popek and Moulton also do not disclose the additional limitation on the "core replicas" that is recited in claim 12. While Yu discusses replication scale, Yu does not teach or suggest applying such a replication scale to "core replicas" connected in the manner recited in Applicants' claim 12.

Therefore, the Applicants respectfully submit that this is yet another reason why claim 12 is allowable.

e. Rejections under 35 U.S.C. § 103 in view of Popek, Moulton and Coram

Claim 13

Claim 13 is dependent from claim and recites "wherein a replica of an object is deleted by marking the replica as invalid." The final rejection alleges that Coram discloses this feature at paragraph [0047] and that it would have been obvious to combine this feature into Popek and Moulton. See, final rejection mailed on July 31, 2007, at pages 15-16.

The Applicants respectfully disagree. Coram is directed toward result set caching in connection with an informational database. See, Abstract of Coram. Therefore, Coram is completely unrelated to distributed file systems, and to replication in distributed file systems. Therefore, there would not have been a motivation to combine Coram into the other references in a manner that would achieve the Applicants' claimed invention. Further, because Popek teaches that concepts taken from the field of databases and transaction-based data storage systems are not applicable to distributed file systems, and Coram describes just such a system, Popek teaches away from making a combination of Coram with Popek. For at least this reason, the Applicants respectfully request reversal of the rejection of claim 13.

Moreover, paragraph [0047] of Coram, which is relied upon for this rejection, discusses invalidating data in cache memory. This is unrelated to the limitations of Applicants' claim 13 which are directed toward invalidating replicas of objects in a distributed file system wherein the objects comprise files and file directories.

In view of the above, the Applicants respectfully request reversal of the rejection of Applicants' claim 13.

Claim 15

Claims 15 stands or falls together with claim 13 from which it depends.

f. *Rejections under 35 U.S.C. § 103 in view of Popek, Moulton, Coram and Zhang*

Claim 14

Claim 14 is dependent from claim 13 and recites, "wherein said marking the replica as invalid comprises removing the backpointer for the replica." The final rejection alleges that Zhang discloses this feature at page 3, in the section entitled, "2 Problem Abstraction." The Applicants disagree that this section discloses such a feature. Particulary, Zhang does not discuss that marking a replica as invalid comprises removing a backpointer for the replica. Therefore, the Applicants respectfully submit that this is another reason why Applicants' claim 14 is allowable.

Claim 67

Claim 67 is dependent from claim and recites "wherein a replica of an object is deleted by marking the replica as invalid." The final rejection alleges that Coram discloses this feature at paragraph [0047] and that it would have been obvious to combine this feature into Popek and Moulton. See, final rejection mailed on July 31, 2007, at pages 15-16 and 18.

The Applicants respectfully disagree. Coram is directed toward result set caching in connection with an informational database. See, Abstract of Coram. Therefore,

Coram is completely unrelated to distributed file systems, and to replication in distributed file systems. Therefore, there would not have been a motivation to combine Coram into the other references in a manner that would achieve the Applicants' claimed invention. Further, because Popek teaches that concepts taken from the field of databases and transaction-based data storage systems are not applicable to distributed file systems, and Coram describes just such a system, Popek teaches away from making a combination of Coram with Popek. For at least this reason, the Applicants respectfully request reversal of the rejection of claim 67.

Moreover, paragraph [0047] of Coram, which is relied upon for this rejection, discusses invalidating data in cache memory. This is unrelated to the limitations of Applicants' claim 67 which are directed toward invalidating replicas objects in a distributed file system wherein the objects comprise files and file directories.

In view of the above, the Applicants respectfully request reversal of the rejection of Applicants' claim 67.

Claim 68

Claim 68 is dependent from claim 67 and recites, "wherein said marking the replica as invalid comprises removing the backpointer for the replica." The final rejection alleges that Zhang discloses this feature at page 3, in the section entitled, "2 Problem Abstraction." The Applicants disagree that this section discloses such a feature. Particulary, Zhang does not discuss that marking a replica as invalid comprises removing a backpointer for the replica. Therefore, the Applicants respectfully submit that this is another reason why Applicants' claim 68 is allowable.

Claim 69

Claim 69 stands or falls together with claim 67 from which it depends.

(viii) Conclusion

In view of the above, the Applicant submits that all of the pending claims are allowable over the cited art. Accordingly, the Applicant requests that the rejections be reversed.

Respectfully Submitted,

Dated: March 3, 2008

Derek J. Westberg (Reg. No. 40,872)

(viii) Claims Appendix

1. A method for a wide-area file system, including a plurality of nodes storing replicas of objects, the objects being files and file directories, wherein for each replica of an object at a node, a parent directory for the object is replicated at the node, the method comprising:

propagating an update to a replica of a file directory to other replicas of the file directory via a graph, wherein each replica of the file directory has edges to only a subset of the other replicas such that all the replicas of the file directory are connected via the graph; and

in response to receiving a propagated update to a replica of the file directory at a node, updating the replica for the file directory at the node.

2. The method according to claim 1, wherein each replica of an object has a backpointer including an identification of a parent directory for the object and a name of the object in the parent directory.

3. The method according to claim 2, wherein the parent directories are modified when the backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node.

4. The method according to claim 3, wherein modifying the parent directories occurs only after a delay.

5. The method according to claim 3, wherein multiple modifications to the parent directories at the node are performed according to an order in which corresponding updates occur.

6. The method according to claim 3, wherein a modification is performed at the node and an earlier inconsistent modification is ignored.

7. The method according to claim 2, wherein a directory operation affects the backpointer for the object.

8. The method according to claim 7, wherein the directory operation is selected from a group consisting of rename, link and unlink.

9. The method according to claim 8, wherein when the backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node, further comprising modifying the parent directories to be consistent with the backpointer.

10. The method according to claim 1, wherein the replicas of the file directory include core replicas and non-core replicas, the parent directory for the file directory having edges only to the core replicas of the file directory and each core

replica of the file directory having edges to one or more of the non-core replicas of the file directory.

11. The method according to claim 10, wherein in response to a user accessing an object at a node when no replica of the object exists at the node, the method further comprises steps of forming a non-core replica of the parent directory for the object at the node and forming a non-core replica of the object at the node.

12. The method according to claim 10, wherein a minimum number of core replicas are maintained according to a minimum replication factor.

13. The method according to claim 1, wherein a replica of an object is deleted by marking the replica as invalid.

14. The method according to claim 13, wherein said marking the replica as invalid comprises removing the backpointer for the replica.

15. The method according to claim 13, further comprising periodically removing replicas marked as invalid.

16-55. (Canceled)

56. A system including:

a plurality of nodes that store replicas of objects, the objects being files and file directories,

wherein for each replica of an object at a node, the node stores a replica of a parent directory for the object and a backpointer having an identification of the parent directory for the object, and

wherein each replica of a file directory has edges to only a subset of the other replicas of the file directory such that all the replicas of the file directory are connected via the graph and the nodes are configured to propagate updates to replicas of each file directory to other replicas of the file directory via the graph.

57. The system according to claim 56, wherein in response to receiving a propagated update to a replica of the file directory at a node, the node updates the parent directories for the file at the node.

58. The system according to claim 56, wherein when a backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node, the node modifies the parent directories to be consistent with the backpointer.

59. The system according to claim 58, wherein the node modifies the parent directories to be consistent with the backpointer only after a delay.

60. The system according to claim 58, wherein multiple modifications to the parent directories at the node are performed according to an order in which corresponding updates occur.

61. The system according to claim 58, wherein a modification is performed at the node and an earlier inconsistent modification is ignored.

62. The system according to claim 56, wherein a directory operation is affects the backpointer for the object.

63. The system according to claim 62, wherein the directory operation is selected from a group consisting of rename, link and unlink.

64. The system according to claim 62, wherein when a backpointer for a replica of an object at a node is not consistent with the parent directories for the replica of the object at the node, the node modifies the parent directories to be consistent with the backpointer.

65. The system according to claim 56, wherein the replicas of the file directory include core replicas and non-core replicas, the parent directory for the file directory having edges only to the core replicas of the file directory and each core replica of the file directory having edges to one or more of the non-core replicas of the file directory.

66. The system according to claim 56, wherein in response to a user accessing an object at a node when no replica of the object exists at the node, a non-core replica of the object and a non-core replica of the parent directory for the object are formed at the node.

67. The system according to claim 56, wherein a replica of an object is deleted by marking the replica as invalid.

68. The system according to claim 67, wherein said marking the replica as invalid comprises removing the backpointer for the replica.

69. The system according to claim 67, further comprising periodically removing replicas marked as invalid.

(ix) Evidence Appendix

None.

(x) Related Proceedings Appendix

None